



TITLE:

\mathbb{R}^2 上のファジィ入力を用いたニューラルネットワーク (決定理論とその関連分野)

AUTHOR(S):

菊池, 洋一; 古川, 長太

CITATION:

菊池, 洋一 ...[et al]. \mathbb{R}^2 上のファジィ入力を用いたニューラルネットワーク (決定理論とその関連分野). 数理解析研究所講究録 1998, 1043: 41-49

ISSUE DATE:

1998-04

URL:

<http://hdl.handle.net/2433/62133>

RIGHT:

R^2 上のファジィ入力を用いたニューラルネットワーク

創価大学工学部 菊池 洋一 (Yoichi Kikuchi)

創価大学工学部 古川 長太 (Nagata Furukawa)

1. はじめに

実ベクトルを入力値とするニューラルネットにおける2群識別問題を考えるとき、線形分離可能な場合は識別可能である。また、線形分離不可能な場合でも、2つの群が互いの領域をさほど侵略していない場合ならば識別可能である。一般的に、データの属性値にあいまいさを含むことは十分に考えられる。このような場合において、近年入力値として実数値からファジィ数へ拡張した研究がなされている。代表的な研究としては、石淵久生先生らによる、ファジィ数を成分とするような実ベクトルを入力とする場合についての2群識別問題がある。

本研究ではファジィ数を成分とするような実ベクトルではなくて R^2 上のファジィ数を入力とする2群識別問題を取り上げる。ここでは、 R^2 上のファジィ数として特にピラミッド型ファジィ数という概念を導入しそれによって研究を行った。そして、ニューラルネットにおいて学習がどの程度まで可能であるかを検討する。

2. ファジィ数

ピラミッド型ファジィ数 A を (m, n) をピラミッドの中心座標、 a, b をそれぞれ x 軸方向、 y 軸方向の幅とした時、 A を

$$A = (m, n, a, b) \quad (1)$$

と表記する。メンバーシップ関数は次の式で与えられる。

$$\mu_A(x, y) = \left(1 - \max \left[\frac{|x - m|}{a}, \frac{|y - n|}{b} \right] \right) \wedge 0, \quad x, y \in R \quad a, b > 0 \quad (2)$$

ここで、(2)式を h とおくと、 h レベル で切った閉矩形

$$[x^L, x^U] \times [y^L, y^U] \quad (3)$$

が求まる。

ファジィ数をニューラルネットに入力する時、実際には各レベル集合における矩形入力を用いる。矩形入力における演算を行う際に、本研究ではファジィ拡張原理により次のようなファジィ演算を行う。

(1) 加法

$$(m, n, a, b) \oplus (o, p, c, d) = (m + o, n + p, a + c, b + d) \quad (4)$$

(2) 負数

$$-(m, n, a, b) = (-m, -n, a, b) \quad (5)$$

3. ニューラルネットの構築

本研究では次のような3層からなる階層型ニューラルネットを用いている。

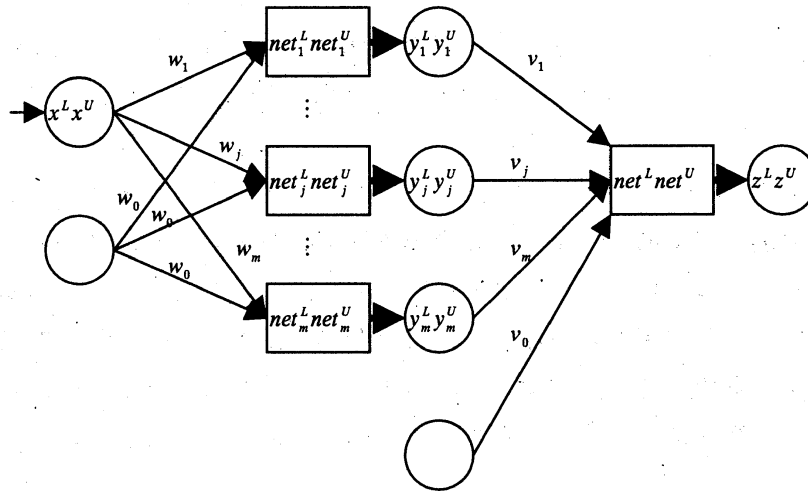


図 3 - 1

$$x^L = \begin{pmatrix} x^{1L} \\ x^{2L} \end{pmatrix} \quad x^U = \begin{pmatrix} x^{1U} \\ x^{2U} \end{pmatrix} \quad (6)$$

$$y_j^L = \begin{pmatrix} y_j^{1L} \\ y_j^{2L} \end{pmatrix} \quad y_j^U = \begin{pmatrix} y_j^{1U} \\ y_j^{2U} \end{pmatrix} \quad (7)$$

$$z^L = \begin{pmatrix} z^{1L} \\ z^{2L} \end{pmatrix} \quad z^U = \begin{pmatrix} z^{1U} \\ z^{2U} \end{pmatrix} \quad (8)$$

(6)(7)(8)式はニューラルネットへの矩形入力を表している。これを図で表現すると次のようになる。

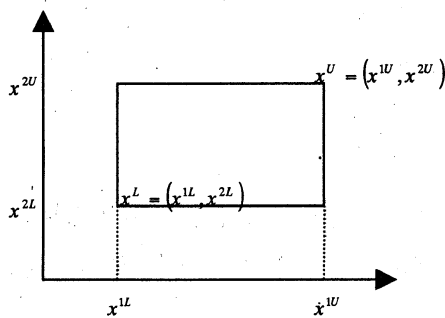


図 3 - 2

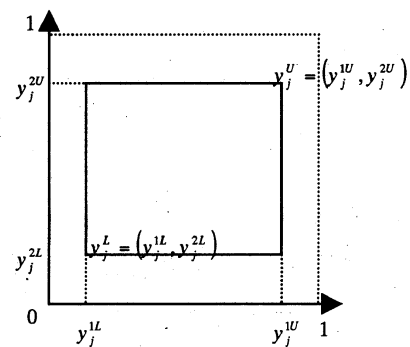


図 3 - 3

図 3 - 2, 図 3 - 3 のように, ニューラルネットに矩形を入力する際, 教師信号である 0, 1 から最も遠い点と最も近い点の 2 点を入力すれば良い。これは矩形の他の 2 点, 最も遠い点と最も近い点によって一意に決まるからである。

各層の各素子における演算式は次のように表される.

$$\delta^+(t) = \begin{cases} 1 & \text{if } t > 0 \\ 0 & \text{if } t \leq 0 \end{cases} \quad (9)$$

$$\delta^-(t) = \begin{cases} 1 & \text{if } t < 0 \\ 0 & \text{if } t \geq 0 \end{cases} \quad (10)$$

入力層 :

$$x^L = \begin{pmatrix} x^{1L} \\ x^{2L} \end{pmatrix} \quad x^U = \begin{pmatrix} x^{1U} \\ x^{2U} \end{pmatrix} \quad (11)$$

中間層 :

$$\begin{aligned} net_j^L &= \begin{pmatrix} net_j^{1L} \\ net_j^{2L} \end{pmatrix} = \delta^+(w_j)w_jx^L + \delta^-(w_j)w_jx^U + w_0 \\ &= \begin{pmatrix} \delta^+(w_j)w_jx^{1L} + \delta^-(w_j)w_jx^{1U} + w_0 \\ \delta^+(w_j)w_jx^{2L} + \delta^-(w_j)w_jx^{2U} + w_0 \end{pmatrix} \end{aligned} \quad (12)$$

$$\begin{aligned} net_j^U &= \begin{pmatrix} net_j^{1U} \\ net_j^{2U} \end{pmatrix} = \delta^+(w_j)w_jx^U + \delta^-(w_j)w_jx^L + w_0 \\ &= \begin{pmatrix} \delta^+(w_j)w_jx^{1U} + \delta^-(w_j)w_jx^{1L} + w_0 \\ \delta^+(w_j)w_jx^{2U} + \delta^-(w_j)w_jx^{2L} + w_0 \end{pmatrix} \end{aligned} \quad (13)$$

$$y_j^L = \begin{pmatrix} y_j^{1L} \\ y_j^{2L} \end{pmatrix} = f(net_j^{1L}, net_j^{2L}) \quad y_j^U = \begin{pmatrix} y_j^{1U} \\ y_j^{2U} \end{pmatrix} = f(net_j^{1U}, net_j^{2U}) \quad (14)$$

出力層 :

$$\begin{aligned} net^L &= \begin{pmatrix} net^{1L} \\ net^{2L} \end{pmatrix} = \sum_{j=1}^m \delta^+(v_j)v_jy_j^L + \sum_{j=1}^m \delta^-(v_j)v_jy_j^U + v_0 \\ &= \begin{pmatrix} \sum_{j=1}^m \delta^+(v_j)v_jy_j^{1L} + \sum_{j=1}^m \delta^-(v_j)v_jy_j^{1U} + v_0 \\ \sum_{j=1}^m \delta^+(v_j)v_jy_j^{2L} + \sum_{j=1}^m \delta^-(v_j)v_jy_j^{2U} + v_0 \end{pmatrix} \end{aligned} \quad (15)$$

$$\begin{aligned}
 \begin{pmatrix} net^{1U} \\ net^{2U} \end{pmatrix} &= \sum_{j=1}^m \delta^+(v_j) \nu_j y_j^U + \sum_{j=1}^m \delta^-(v_j) \nu_j y_j^L + v_0 \\
 &= \begin{pmatrix} \sum_{j=1}^m \delta^+(v_j) \nu_j y_j^{1U} + \sum_{j=1}^m \delta^-(v_j) \nu_j y_j^{1L} + v_0 \\ \sum_{j=1}^m \delta^+(v_j) \nu_j y_j^{2U} + \sum_{j=1}^m \delta^-(v_j) \nu_j y_j^{2L} + v_0 \end{pmatrix}
 \end{aligned} \tag{16}$$

$$z^L = \begin{pmatrix} z^{1L} \\ z^{2L} \end{pmatrix} = f(net^{1L}, net^{2L}) \quad z^U = \begin{pmatrix} z^{1U} \\ z^{2U} \end{pmatrix} = f(net^{1U}, net^{2U}) \tag{17}$$

4. シグモイド関数

入力層—中間層間, 中間層—出力層間において, 各素子の入出力関係はシグモイド関数と呼ばれる関数によって定義される. 本研究においては, シグモイド関数は0と1の間の値を取り, 次のように表される.

$$f(x, y) = (\tau(x), \tau(y)) \tag{18}$$

ここで,

$$\tau(s) = \frac{1}{1 + e^{-s}} \tag{19}$$

である. よって,

$$f(A \times B) = C \times D \subset [0, 1] \times [0, 1] \tag{20}$$

となる. シグモイド関数は次のようなグラフで表される.

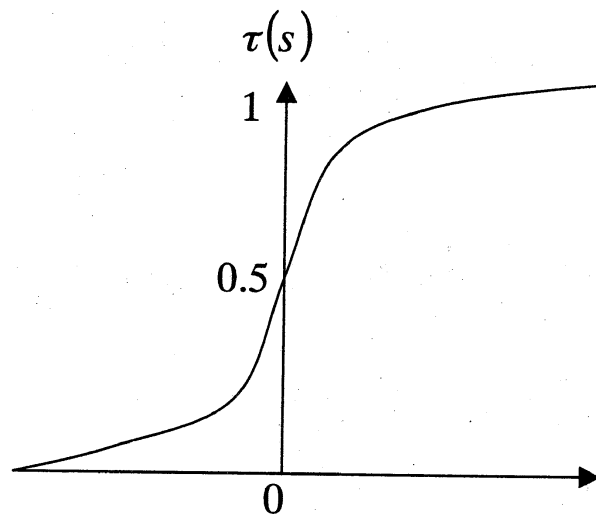


図4—1

5. 学習アルゴリズム

5-1. 誤差関数

h レベルで切った時の誤差関数

$$t_p = \begin{pmatrix} 0.9 \\ 0.9 \end{pmatrix} \text{ のとき } E^h = \left[(z^{1L} - 0.9)^2 + (z^{2L} - 0.9)^2 + (z^{1U} - 0.9)^2 + (z^{2U} - 0.9)^2 \right] / 2 \quad (21)$$

$$t_p = \begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix} \text{ のとき } E^h = \left[(z^{1L} - 0.1)^2 + (z^{2L} - 0.1)^2 + (z^{1U} - 0.1)^2 + (z^{2U} - 0.1)^2 \right] / 2 \quad (22)$$

ピラミッド型ファジィ数を与えた場合の誤差関数

h カットを 0, 0.2, 0.4, 0.6, 0.8 でそれぞれ行い, 各レベルでの誤差に h レベルで重み付けをして, その和を誤差関数とします. ただし, 0 カットでの誤差への重みは 0.05 とします.

$$E = \sum h E^h \quad (23)$$

$$E = 0.05 E^0 + 0.2 E^{0.2} + 0.4 E^{0.4} + 0.6 E^{0.6} + 0.8 E^{0.8} \quad (24)$$

これを図を使って説明すると, 各教師信号において, 矩形の最も遠い点と最も近い点との誤差を 2 乗して足したものを意味している.

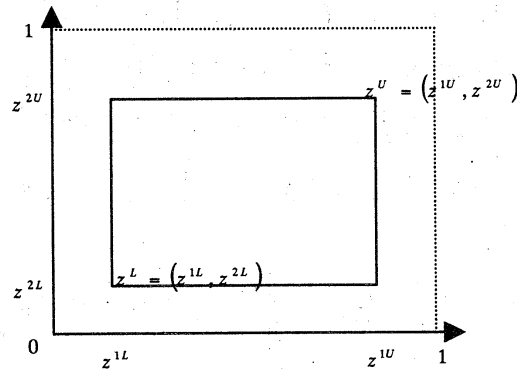


図 5-1

5-2. 結合重みの修正

ニューラルネットの学習を意味することは, 誤差関数 E を最小にするように結合重みを修正することである. 普通, ニューラルネットでは誤差関数の最小化法として最急降下法が用いられている. 最急降下法により結合重みは次のように修正される.

$$\begin{pmatrix} w^{k+1} \\ v^{k+1} \end{pmatrix} = \begin{pmatrix} w^k \\ v^k \end{pmatrix} + (\beta)^T E(w^k, v^k) \quad (25)$$

入力層—中間層間の重みを w_j とし, 中間層—出力層間の重みを v_j としたとき $\partial E / \partial w_j$, $\partial E / \partial v_j$ は次のように導出される. $j=1, 2, \dots, m$ に対して,

$$\square \quad w_j \geq 0, v_j \geq 0$$

$$\frac{\partial E}{\partial w_j} = -v_j \begin{bmatrix} z^{1L}(t_p - z^{1L})(1 - z^{1L})y_j^{1L}(1 - y_j^{1L})x^{1L} \\ + z^{2L}(t_p - z^{2L})(1 - z^{2L})y_j^{2L}(1 - y_j^{2L})x^{2L} \\ + z^{1U}(t_p - z^{1U})(1 - z^{1U})y_j^{1U}(1 - y_j^{1U})x^{1U} \\ + z^{2U}(t_p - z^{2U})(1 - z^{2U})y_j^{2U}(1 - y_j^{2U})x^{2U} \end{bmatrix} \quad (26)$$

$$\square \quad w_j \geq 0, v_j < 0$$

$$\frac{\partial E}{\partial w_j} = -v_j \begin{bmatrix} z^{1L}(t_p - z^{1L})(1 - z^{1L})y_j^{1U}(1 - y_j^{1U})x^{1U} \\ + z^{2L}(t_p - z^{2L})(1 - z^{2L})y_j^{2U}(1 - y_j^{2U})x^{2U} \\ + z^{1U}(t_p - z^{1U})(1 - z^{1U})y_j^{1L}(1 - y_j^{1L})x^{1L} \\ + z^{2U}(t_p - z^{2U})(1 - z^{2U})y_j^{2L}(1 - y_j^{2L})x^{2L} \end{bmatrix} \quad (27)$$

$$\square \quad w_j < 0, v_j \geq 0$$

$$\frac{\partial E}{\partial w_j} = -v_j \begin{bmatrix} z^{1L}(t_p - z^{1L})(1 - z^{1L})y_j^{1L}(1 - y_j^{1L})x^{1U} \\ + z^{2L}(t_p - z^{2L})(1 - z^{2L})y_j^{2L}(1 - y_j^{2L})x^{2U} \\ + z^{1U}(t_p - z^{1U})(1 - z^{1U})y_j^{1U}(1 - y_j^{1U})x^{1L} \\ + z^{2U}(t_p - z^{2U})(1 - z^{2U})y_j^{2U}(1 - y_j^{2U})x^{2L} \end{bmatrix} \quad (28)$$

$$\square \quad w_j < 0, v_j < 0$$

$$\frac{\partial E}{\partial w_j} = -v_j \begin{bmatrix} z^{1L}(t_p - z^{1L})(1 - z^{1L})y_j^{1U}(1 - y_j^{1U})x^{1L} \\ + z^{2L}(t_p - z^{2L})(1 - z^{2L})y_j^{2U}(1 - y_j^{2U})x^{2L} \\ + z^{1U}(t_p - z^{1U})(1 - z^{1U})y_j^{1L}(1 - y_j^{1L})x^{1U} \\ + z^{2U}(t_p - z^{2U})(1 - z^{2U})y_j^{2L}(1 - y_j^{2L})x^{2U} \end{bmatrix} \quad (29)$$

$$\begin{aligned} \frac{\partial E}{\partial w_0} = & -z^{1L}(t_p - z^{1L})(1 - z^{1L}) \left[\sum_{v_j \geq 0} v_j y_j^{1L}(1 - y_j^{1L}) + \sum_{v_j < 0} v_j y_j^{1U}(1 - y_j^{1U}) \right] \\ & - z^{2L}(t_p - z^{2L})(1 - z^{2L}) \left[\sum_{v_j \geq 0} v_j y_j^{2L}(1 - y_j^{2L}) + \sum_{v_j < 0} v_j y_j^{2U}(1 - y_j^{2U}) \right] \\ & - z^{1U}(t_p - z^{1U})(1 - z^{1U}) \left[\sum_{v_j \geq 0} v_j y_j^{1U}(1 - y_j^{1U}) + \sum_{v_j < 0} v_j y_j^{1L}(1 - y_j^{1L}) \right] \\ & - z^{2U}(t_p - z^{2U})(1 - z^{2U}) \left[\sum_{v_j \geq 0} v_j y_j^{2U}(1 - y_j^{2U}) + \sum_{v_j < 0} v_j y_j^{2L}(1 - y_j^{2L}) \right] \quad (30) \end{aligned}$$

$$\square \quad v_j \geq 0 \quad \frac{\partial E}{\partial v_j} = -z^{1L}(t_p - z^{1L})(1 - z^{1L})y_j^{1L} - z^{2L}(t_p - z^{2L})(1 - z^{2L})y_j^{2L} \\ - z^{1U}(t_p - z^{1U})(1 - z^{1U})y_j^{1U} - z^{2U}(t_p - z^{2U})(1 - z^{2U})y_j^{2U} \quad (31)$$

$$\square \quad v_j < 0 \quad \frac{\partial E}{\partial v_j} = -z^{1L}(t_p - z^{1L})(1 - z^{1L})y_j^{1U} - z^{2L}(t_p - z^{2L})(1 - z^{2L})y_j^{2U} \\ - z^{1U}(t_p - z^{1U})(1 - z^{1U})y_j^{1L} - z^{2U}(t_p - z^{2U})(1 - z^{2U})y_j^{2L} \quad (32)$$

$$\frac{\partial E}{\partial v_0} = -z^{1L}(t_p - z^{1L})(1 - z^{1L}) - z^{2L}(t_p - z^{2L})(1 - z^{2L}) \\ - z^{1U}(t_p - z^{1U})(1 - z^{1U}) - z^{2U}(t_p - z^{2U})(1 - z^{2U}) \quad (33)$$

5-3. アーミホの方法

$j=1 \cdots m$ で, ある重み w_j , v_j において

$$\nabla E(w_j, v_j) s < 0 \quad (34)$$

なるベクトル s が求まったとする. そして, p を $0, 1, 2, \dots$ と変えていって,

$$E(w_j + (\beta)^p, v_j + (\beta)^p) - E(w_j, v_j) \leq (\beta)^p \mu \nabla E(w_j, v_j) s < 0 \quad (35)$$

が初めて成り立ったときの p の値を l とする. このときの β^l を s 方向のステップ幅として採用する方法をアーミホの方法という.

アーミホの方法で μ , β は通常,

$$10^{-4} \leq \mu \leq 10^{-1}, \quad 10^{-1} \leq \beta \leq \frac{1}{2} \quad (36)$$

ととられる.

6. 学習結果

実際に次のような数値例をニューラルネットに学習させた結果 p8 に示すグラフィックによる実行結果を得た. 破線で表されるピラミッドの群が教師信号の 0 に当たり, 実線で表させる群が教師信号の 1 に当たる.

数値例:

数値 教師信号 '0'

(2.5, 1.5, 0.5, 0.5)

(0.5, 2.5, 0.5, 0.5)

(4.0, 3.2, 1.0, 0.5)

(3.3, 4.4, 0.5, 0.6)

(0.5, 0.5, 0.5, 0.5)

(2.0, 3.2, 0.5, 0.8)

数値 教師信号 '1'

(7.5, 4.5, 0.5, 0.5)

(4.8, 3.8, 0.5, 0.5)

(4.2, 5.3, 0.7, 0.7)

(5.5, 7.5, 0.5, 0.5)

(2.5, 7.0, 1.0, 0.5)

(5.5, 1.5, 0.5, 0.5)

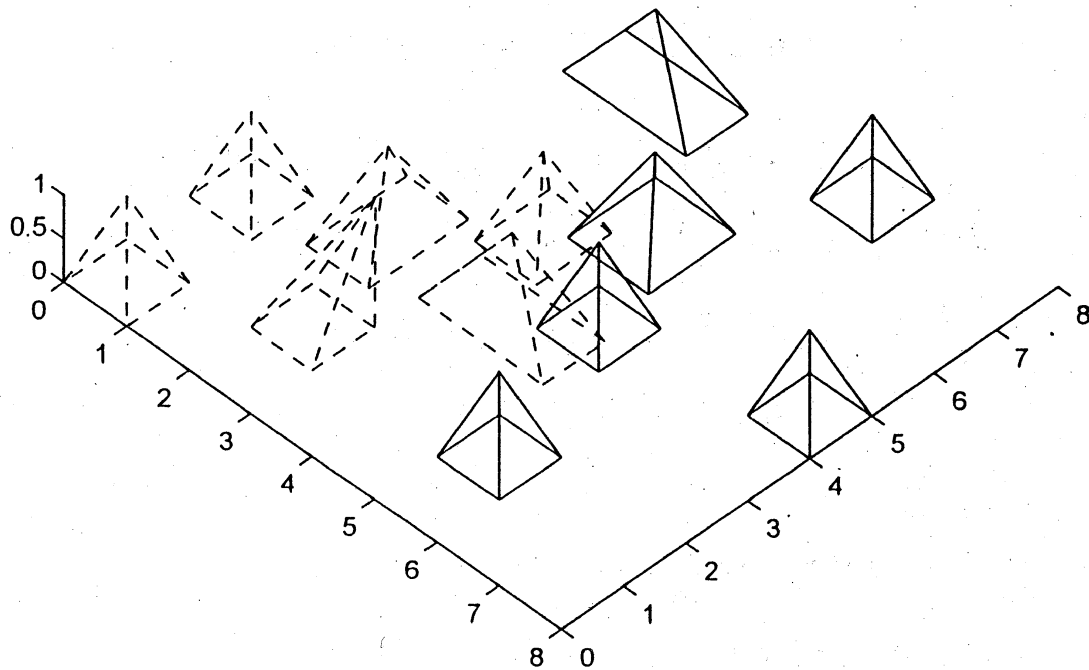


図6-1

7. まとめ

7-1. 学習の考察

データパターンを色々と変えてシミュレーションしてみた結果、ある程度視覚的に分かれている状態ならば、学習が可能であることが結果として得られた。特に x , y と共に線形分離可能ならば学習速度も速く、学習度合いも非常に高い様子が見られた。しかし、ピラミッドの重なり度合いが大きかったり、完全に重なっていたり、大きいピラミッドに小さいピラミッドが含まれてしまうような場合には、学習が困難になることが分かった。また、誤差関数において遠い点と近い点を足して2で割らないで、遠い点だけからの距離だけを考えた誤差関数では学習はあまりうまくいかなかった。そして、重みの修正の際に共役勾配法を導入することは、前の重みの修正量の影響を受けるので、ファジィ数を扱う場合にはその効果を期待できないことが分かった。

7-2. 結び

本研究ではニューラルネットにおける2群識別問題において、従来の三角型ファジィ数からピラミッド型ファジィ数への拡張の方法を示した。2次元データを取り扱う場合に、本研究の方法は有効であると言える。今後の研究課題としては、実際の応用例に当てはめてみて、その有用性を示していくことがあげられる。また、ファジィ数の型を円錐や墮円錐のような場合に拡張して、2群識別問題を考えることなどもあげられる。

参考文献

- [1] 研究集会：「不確実性の下における意思決定のための数理モデルとその応用」千葉大学(1997), 古川長太：「 R^2 におけるコンパクト凸集合間の順序関係と R^2 上のファジィ数間の順序関係について」
- [2] 石淵久夫, 岡田英彦, 田中英夫：「ファジィ入出力データを用いたニューラルネットの学習」, 日本ファジィ学会誌 Vol.4, No.5, pp.892-905(1992)
- [3] 石淵久夫, 藤岡亮介, 田中英夫：「ニューラルネットによるファジィ数ベクトルの識別」, システム制御情報学会誌, Vol.5, No.5, pp.198-206(1992)
- [4] 石淵久夫：「ファジィ入力とファジィ出力をもつニューラルネットワーク」, 日本ファジィ学会誌, Vol.5, No.2, pp.218-232(1993)
- [5] 相吉英太郎, 志水清孝：数理計画法演習, 朝倉書店
- [6] 上坂喜則：ニューロコンピューティングの数学的基礎, 近代科学社
- [7] 水元雅晴：ファジィ理論とその応用